LEARNING AND TRANSFERRING REPRESENTATIONS FOR IMAGE STEGANALYSIS USING CONVOLUTIONAL NEURAL NETWORK

Yinlong Qian[1,2], *Jing Dong*[2,3*], *Wei Wang*[2] *and Tieniu Tan*[2]

[1]Department of Automation,University of Science and Technology of China
[2]Center for Research on Intelligent Perception and Computing,
Institute of Automation, Chinese Academy of Sciences,
[3]State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences
E-mail: ylqian@mail.ustc.edu.cn, {jdong, wwang, tnt}@nlpr.ia.ac.cn

ABSTRACT

The major challenge of machine learning based image steganalysis lies in obtaining powerful feature representations. Recently, Qian et al. have shown that Convolutional Neural Network (CNN) is effective for learning features automatically for steganalysis. In this paper, we follow up this new paradigm in steganalysis, and propose a framework based on transfer learning to help the training of CNN for steganalysis, hence to achieve a better performance. We show that feature representations learned with a pre-trained CNN for detecting a steganographic algorithm with a high payload can be efficiently transferred to improve the learning of features for detecting the same steganographic algorithm with a low payload. By detecting representative WOW and S-UNIWARD steganographic algorithms, we demonstrate that the proposed scheme is effective in improving the feature learning in CNN models for steganalysis.

Index Terms— Steganalysis, deep learning, transfer learning, Convolutional Neural Network

1. INTRODUCTION

The goal of image steganalysis is to detect the presence of secret messages in digital images. Usually, this task is viewed as a binary classification problem to distinguish between covers and stegos. Most of existing steganalysis approaches follow a conventional paradigm based on machine learning, which consists of two steps. The first step extracts features from images, and the second step train a classifier, such as SVM and the FLD-based ensemble classifier [1], based on the extracted features. The major challenge lies in extracting effective image representations to capture enough traces caused by embedding operations. In the past decades, researchers have focused on designing appropriate feature extractors, and various handcrafted features have been proposed in the literature [2–10]. Though significant progress has been achieved in recent years, the detection accuracy of existing steganalysis systems based on handcrafted features is far from satisfactory. Moreover, the handcrafted feature design is heavily dependent on expert experiences, and it is difficult and time-consuming to design new features.

In recent years, deep learning, which addresses the problem of what makes better feature representations and how to learn them from data, is becoming a hot topic in the field of machine learning, and has shown competitive performance in learning effective representations for many tasks. Qian et al. [11] showed that the use of Convolutional Neural Network (CNN) [12], one of the most representative deep learning models, to automatically learn effective features is very promising for the steganalysis task. Compared with traditional steganalysis systems, this feature learning based approach unifies feature extraction and classification modules under a single network architecture, and jointly optimizes all the parameters in both modules with a learning algorithm. It can be viewed as a step towards automatizing steganalysis detectors which require minimal domain knowledge and little human labor.

In this paper, we move a step forward, and propose to use transfer learning to help the training of CNN based model for steganalysis, hence to achieve a better performance. The core idea of our approach is that feature representations learned with CNN for the task of detecting a steganographic algorithm with a high payload can be efficiently transferred to improve the learning of features for the task of detecting the same steganographic algorithm with a lower payload. In fact, though CNN has been shown to be effective for the steganalysis problem, we observed that the lower the payload for a steganographic algorithm, the harder the training of C-

*Corresponding author. Dr. Wei Wang is also a visiting researcher with Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China.

This work is funded by the National Nature Science Foundation of China (Grant No.61303262, No.U1536120 and No.61502496) and Beijing Natural Science Foundation (Grant No.4164102).

NN. What's worse, CNN may even fail to converge when the payload is rather low. This is mainly due to the fact that embedding less message bits leaves less traces, which makes the obtained stego image much harder to detect. To address this problem, in our work, we incorporate auxiliary information from stegos obtained using a steganographic algorithm with a high payload through transfer learning to improve the training of CNN for detecting the same steganographic algorithm with a lower payload. This idea is much different from the traditional concept of steganalysis that the tasks of detecting a steganographic algorithm with different payloads are usually treated independently. However, in this paper, we argue that these tasks are subtly correlated. We believe that there are some crucial patterns for steganalysis caused by embedding operations shared between stegos with different payloads. Moreover, these patterns are more significant and easier to learn in stegos with a higher payload, and can be efficiently transferred to improve the performance for the task of detecting stegos with a lower payload.

2. RELATED WORK

Most of traditional steganalysis schemes rely on handcrafted features, and the efforts are mainly devoted to the design of powerful feature descriptors by hand. The current state-of-the-art features are some high dimensional vectors called rich representations [2, 4–7], which are designed to capture complex statistics such as dependencies among neighboring pixels. These rich representations are generated by assembling a number of submodels formed from noise residuals obtained using linear and non-linear high-pass filters. Different from traditional steganalysis approaches, our work focuses on learning effective features automatically based on deep learning.

In previous work [11], we propose to use deep learning for steganalysis to replace the traditional two step approach. The proposed framework is based on Convolutional Neural Network, one of the most representative deep learning models. It first uses a fixed high-pass filter for preprocessing, and then extracts feature representations with multiple layers of convolution and pooling operations, and finally passed the extracted features to fully connected layers for classification. In a word, feature extraction and classification modules are unified under a single network architecture. The whole network is trained using the back-propagation algorithm to jointly optimize all the parameters in both steps. Differently to this work, we here consider the use of transfer learning to help the training of CNN model for achieving a better performance for steganalysis.

Generally, the goal of transfer learning is to leverage shared domain-specific knowledge contained in related tasks to help improving the performance of the target task. In [13–15], transfer learning with CNNs is explored for visual recognition in a manner of joint training CNNs from unsupervised pseudo-tasks. Differently to these approaches, we pre-train CNNs on a supervised task of detecting stegos with a high payload, and then transfer the CNN parameters for the task of detecting stegos with a lower payload. Transfer learning with CNNs in a similar manner of reusing layers of a pre-trained CNN has been also explored for object recognition [16] and video classification [17].

3. PROPOSED FRAMEWORK

In this section, we introduce the proposed framework in detail. The proposed framework is illustrated in Fig. 1, showing that the feature representations obtained from a pre-trained CNN for detecting stegos with a high payload A (the source task) are transferred to help the training of CNN for detecting stegos with a lower payload B (the target task).

3.1. CNN based model

For both the source task and the target task, we use the network architecture of Qian et al. [11]. The network consists of one image processing layer P1, five convolutional layers C1, C2, C3, C4, C5 as the feature extraction module, and three fully connected layers F1, F2, F3 as the classification module (see Fig. 1, top).

The image processing layer filters the input image with the fixed high-pass KV filter kernel of size 5×5 (see Eqn. (1)) to obtain a noise residual. Note that, this hardwired layer is important and specific to the steganalysis problem, and CNN does not converge without the mandatory high-pass filtering operation.

$$K_{kv} = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \quad (1)$$

At each of the five convolutional layers, convolution, non-linear activation, and pooling operations are applied sequentially, generating 16 feature maps. Mathematically, the three consecutive operations can be summarized as below.

$$X_j^l = pool(f(\sum_i X_i^{l-1} * K_{ij}^l + b_j^l)), \quad (2)$$

where $f()$ denotes non-linearity operation, $pool()$ denotes pooling, X_j^l is the j-th feature map in layer l, X_i^{l-1} is the i-th feature map in layer $l-1$, K_{ij} is the trainable filter connecting the j-th output map and the i-th input map, b_j^l is an trainable bias parameter for the j-th output map. Note that, the weights of filter kernels and the biases have to be learned and are modified during training.

The first convolutional layer accepts the noise residual from the image processing layer as input and filters it with

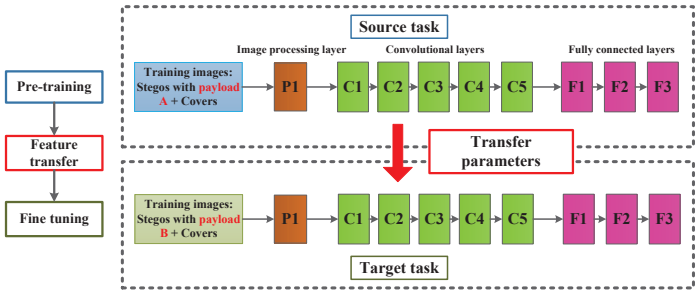

Fig. 1. Flowchart of the proposed framework.

16 trainable kernels of size 5×5. The second, third and fourth convolutional layers apply convolutions with the kernel size of 3×3. The size of convolution kernel used in the fifth convolutional layer is 5×5. The filtering stride of all convolution operations in the five convolutional layers is 1. At each convolutional layer, the Gaussian activation function is applied element-wise to the output of convolution operations. Moreover, each of the five convolutional layers applies an overlapping average pooling operation with the window size 3×3 and stride 2.

After five layers of convolution and pooling operations, the input image has been converted into a 256D feature vector capturing the steganographic traces in the input image, and are finally fed to the classification module consisting of three fully connected layers. Each of the first two fully connected layers have 128 neurons, and the output of each neuron is activated by the ReLU activation function [18]. The last fully connected layer has 2 neurons, and the outputs are fed to a two-way softmax for classification.

3.2. Learning and transferring features

Based on the described network architecture, here we introduce how features can be learned from the source task and transferred to the target task. First, we pre-train the network on the task of detecting stegos with a higher payload A (the source task) using the back-propagation algorithm. For the source task, the training images are composed of stegos with a higher payload A and the corresponding covers. Note that, the KV kernel in the image processing layer is fixed , while all the trainable parameters in the network are initialized randomly and learned during training. The trainable parameters here include filter kernels and the biases in the convolutional layers, as well as weights and biases in the fully connected layers.

After pre-training the network on the source task, we transfer the parameters of the five convolution layers C1, C2, C3, C4, C5 and three fully connected layers, F1, F2, F3 to the target task of detecting stegos with a lower payload B, that is, we initialize the network for the target task with the parameters learned from the source task. Then we fine tune the network using training images consisting of stegos with a lower payload B and the corresponding covers. Note that, the pre-training and fine-tuning procedures are similar, except that the former initializes the trainable parameters randomly, while the latter initializes the network with the representations already learned from the pre-trained network. But the initialization from the pre-trained network indeed plays an important role in improving the training of CNN on the target task. In fact, though CNN has shown great discriminative power in many image classification tasks, it is prone to getting stuck in local minima, which is a common weakness of neural networks, especially deep networks. For the steganalysis task, when embedding with a very low payload, the differences between stegos and covers are quite small, which makes CNN hard to train. However, the shared representations of the pre-trained network on the source task of detecting stegos with a much higher payload already capture some important patterns caused by embedding operations, hence providing a good regularization to drive the network training for the target task.

4. EXPERIMENTS

In this paper, all experiments were carried out on the standardized BOSSbase 1.01 dataset [19] containing 10,000 cover images of size 512×512. We split the dataset by assigning 70% of the images to a training set, 10% to a validation set, and 20% to a test set, respectively. It is necessary to point out that, we use the same split for all the experiments.

Due to the GPU memory limitation, it is hard for our proposed deep network to directly use an image of size 512×512 as the input. In our experiments, we tackle this problem by firstly extracting five 256×256 patches, including the four corner patches and the center patch, and their flip version from each image of size 512×512 to represent the whole image, and then feed these extracted patches to the CNN network described above. At test time, we first make a prediction on each of ten patches extracted from an image of size 512×512, and then average the ten predictions to produce a estimate of the class probabilities for the entire image. This

transformation can greatly reduce the GPU memory requirement, while artificially enlarging the dataset to reduce the effects of overfitting.

The training of our proposed network was carried out using the code provided by Krizhevsky et al [20]. We use mini-batch size of 128 and momentum of 0.9. The weight decay is 0 for the convolutional layers and 0.01 for the fully connected layers. All models are initialized with learning rates of 0.001. The training is stopped whenever the validation error stops improving. In our experiments, the number of iterations is 100 to 200 for pre-training. During the fine-tuning, we first train the pre-trained model for 10 to 20 iterations, then divide the learning rates by 10 and train the model for another 10 to 20 iterations.

We evaluate the performance of the proposed approach on detecting WOW [21] and S-UNIWARD [22], two of the state-of-the-art spatial domain steganographic algorithms, across five payloads 0.1, 0.2, 0.3, 0.4, and 0.5 bpp (bits per pixel). For each of the two steganographic algorithms, we first train a CNN network on stegos with a payload A and the corresponding covers, and then apply our transfer learning scheme to the tasks of detecting the same steganographic algorithm with a payload lower than A. For example, the "Pre-0.6bpp" means that we first pre-train the network on stegos with the payload 0.6 bpp and the corresponding covers, and then transfer the learned parameters to the task of detecting stegos with the payload of 0.5, 0.4, 0.3, 0.2, and 0.1 bpp, respectively. We compared our method with the CNN model without transfer learning ("No-pretrain"), that is the framework proposed by Qian et al. [11], and also with one of the state-of-the-art traditional steganalysis schemes based on handcrafted features, that is the SRM feature set [4] implemented with the Ensemble Classifier ("SRM + EC").

We report the detection error in Table 1 and Table 2. Here the detection error is computed as follow.

$$P_E = min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD}(P_{FA})). \quad (3)$$

The comparison between our proposed method and the framework proposed by Qian et al. in [11] shows that using this transfer learning scheme we have obtained significant improvements in detecting stegos with a payload lower than 0.3 bpp for WOW algorithm, and lower than 0.4 bpp for S-UNIWARD algorithm. We found that, CNN without pre-training does not converge when the payload is as low as 0.1 bpp in our experiments, and the detection error is 50%. But with our transfer learning scheme, we do make the CNN converge and obtain a much better performance. We also observed that the best choice of payload for obtaining stegos in the pre-training step is 0.4 bpp when detecting WOW, and 0.5bpp when detecting S-UNIWARD. A choice of either a lower or a higher payload will lead to a significant performance drop as compared with the best choice. In fact, as the payload for the source task goes lower, and it becomes harder for the pre-trained CNN to capture enough shared patterns for transferring. Moreover, both WOW and S-UNIWARD are content-adaptive steganographic algorithms which embed messages into areas of images that are relatively hard to detect. Note that, the payload in the target task is relatively low, which means the messages are more likely to be embedded in noisy or textured areas that are hard to model. As the payload for the source task goes higher, some messages may be embedded in smooth areas that are easier to model, and it is more likely that the pre-trained CNN will capture the patterns in smooth areas, which are much different from patterns in noisy areas, for transferring. Finally, when compared with the traditional "SRM + EC" method, our approach achieves a better performance on detecting WOW algorithm.

Table 1. Detection error for WOW algorithm.

Payload	0.1bpp	0.2bpp	0.3bpp	0.4bpp	0.5bpp
No-pretrain	50.00%	33.30%	27.88%	20.28%	18.50%
Pre-0.6bpp	40.85%	33.55%	28.28%	22.73%	18.55%
Pre-0.5bpp	40.13%	33.18%	27.48%	21.95%	-
Pre-0.4bpp	**38.43%**	**30.78%**	**24.87%**	-	-
Pre-0.3bpp	40.40%	32.67%	-	-	-
Pre-0.2bpp	39.83%	-	-	-	-
SRM + EC	39.77%	31.75%	24.92%	20.67%	16.23%

Table 2. Detection error for S-UNIWARD algorithm.

Payload	0.1bpp	0.2bpp	0.3bpp	0.4bpp	0.5bpp
No-pretrain	50.00%	37.40%	30.60%	24.08%	17.33%
Pre-0.6bpp	43.80%	35.38%	29.78%	23.33%	18.63%
Pre-0.5bpp	**42.93%**	**34.38%**	**28.42%**	**22.05%**	-
Pre-0.4bpp	43.18%	35.78%	29.57%	-	-
Pre-0.3bpp	43.30%	36.50%	-	-	-
Pre-0.2bpp	43.90%	-	-	-	-
SRM + EC	40.25%	32.10%	24.95%	20.55%	16.64%

5. CONCLUSIONS

This paper proposes a novel framework based on transfer learning to improve the learning of features with CNN models for steganalysis. In the proposed framework, we first pre-train a CNN model using training images composed of stegos with a high payload and the corresponding covers, and then transfer the learned feature representations to regularize the CNN model for a better performance in detecting stegos with a lower payload. In this manner, the auxiliary information from stegos with a high payload can be efficiently utilized to help the task of detecting stegos with a low payload. Experimental results show that the proposed framework does bring an improvement as compared with the previous CNN model for steganalysis without using the transfer learning scheme. Our approach also achieves a better performance than the traditional steganalysis scheme that using SRM feature set when detecting the WOW algorithm.

References

[1] Jan Kodovskỳ, Jessica Fridrich, and Vojtěch Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 432–444, 2012.

[2] Gokhan Gul and Fatih Kurugollu, "A new methodology in steganalysis: breaking highly undetectable steganograpy (hugo)," in *Information Hiding*. Springer, 2011, pp. 71–84.

[3] Jessica Fridrich, Jan Kodovskỳ, Vojtěch Holub, and Miroslav Goljan, "Steganalysis of content-adaptive steganography in spatial domain," in *Information Hiding*. Springer, 2011, pp. 102–117.

[4] Jessica Fridrich and Jan Kodovskỳ, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[5] Yun Q. Shi, Patchara Sutthiwan, and Licong Chen, "Textural features for steganalysis," in *Information Hiding*. Springer, 2013, pp. 63–77.

[6] Vojtech Holub and Jessica Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1996–2006, 2013.

[7] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang, "Adaptive steganalysis against wow embedding algorithm," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, 2014, pp. 91–96.

[8] Tomas Denemark, Vahid Sedighi, Vojtěch Holub, Rémi Cogranne, and Jessica Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *2015 National Conference on Parallel Computing Technologies (PARCOMPTECH)*, 2015, pp. 48–53.

[9] Jan Kodovskỳ and Jessica Fridrich, "Steganalysis of jpeg images using rich models," in *IS&T/SPIE Electronic Imaging*, 2012, pp. 83030A–83030A.

[10] Miroslav Goljan, Jessica Fridrich, Rémi Cogranne, et al., "Rich model for steganalysis of color images," in *Parallel Computing Technologies (PARCOMPTECH), 2015 National Conference on*, 2015, pp. 185–190.

[11] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan, "Deep learning for steganalysis via convolutional neural networks," in *IS&T/SPIE Electronic Imaging*, 2015, pp. 94090J–94090J.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[14] Amr Ahmed, Kai Yu, Wei Xu, Yihong Gong, and Eric Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *ECCV*, pp. 69–82. 2008.

[15] Hossein Mobahi, Ronan Collobert, and Jason Weston, "Deep learning from temporal coherence in video," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 737–744.

[16] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[17] A. Karpathy, G. Toderici, S. Shetty, and T. Leung, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[18] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[19] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ, " break our steganographic system: The ins and outs of organizing boss," in *Information Hiding*. Springer, 2011, pp. 59–70.

[20] A. Krizhevsky, "cuda-convnet," 2012, http://code.google.com/p/cuda-convnet/.

[21] Vojtéch Holub and Jessica Fridrich, "Designing steganographic distortion using directional filters.," in *The IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 234–239.

[22] Vojtěch Holub and Jessica Fridrich, "Digital image steganography using universal distortion," in *Proceedings of the first ACM workshop on Information hiding and multimedia security*. ACM, 2013, pp. 59–68.